

# Δομές ελέγχου της Java

## ΕΝΤΟΛΕΣ ΕΠΙΛΟΓΗΣ

### *if-else*

Η εντολή *if-else* έχει δύο μορφές:

```
if (παράσταση) {
    εντολές
}

if (παράσταση) {
    Εντολές
}
else {
    Εντολές
}
```

Παραδείγματα:

```
if(field.size() == 0) {
    System.out.println("The field is empty.");
}
```

```
if(number < 0) {
    reportError();
}
else {
    processNumber(number);
}
```

```
if(n < 0) {
    handleNegative();
}
else if(number == 0) {
    handleZero();
}
else {
    handlePositive();
}
```

### *switch*

Η εντολή *switch* ελέγχει μία συγκεκριμένη τιμή σε σχέση με έναν οποιονδήποτε αριθμό περιπτώσεων. Δύο πιθανά πρότυπα χρήσης είναι τα ακόλουθα:

```
switch (παράσταση) {
    case τιμή: εντολές;
        break;
    case τιμή: εντολές;
        break;
    οι υπόλοιπες περιπτώσεις
    παραλείπονται
    default: εντολές;
        break;
}

switch (expression) {
    case τιμή1:
    case τιμή2:
    case τιμή3:
        εντολές;
        break;
    case τιμή4:
    case τιμή5:
        εντολές;
        break;
    οι υπόλοιπες περιπτώσεις παραλείπονται
    default:
        εντολές;
        break;
}
```

Σημειώσεις:

- Μια εντολή *switch* μπορεί να έχει οποιονδήποτε αριθμό ετικετών *case*.

- Η εντολή *break* χρειάζεται μετά από κάθε περίπτωση, αλλιώς η εκτέλεση ‘συνεχίζει’ με τις εντολές της επόμενης ετικέτας. Η δεύτερη μορφή παραπάνω χρησιμοποιεί αυτό το γεγονός. Σε αυτή την περίπτωση, και οι τρεις πρώτες τιμές θα εκτελέσουν το πρώτο τμήμα εντολές, ενώ οι τιμές τέσσερα και πέντε θα εκτελέσουν το δεύτερο τμήμα εντολές.
- Η περίπτωση *default* είναι προαιρετική. Αν δεν υπάρχει περίπτωση *default*, μπορεί καμία περίπτωση να μην εκτελεστεί.
- Η εντολή *break* μετά από την περίπτωση *default* (ή την τελευταία περίπτωση *case*, αν δεν υπάρχει περίπτωση *default*) δεν χρειάζεται, αλλά θεωρείται καλό στυλ.

Παραδείγματα:

```
switch(day) {
    case 1: dayString = "Monday";
            break;
    case 2: dayString = "Tuesday";
            break;
    case 3: dayString = "Wednesday";
            break;
    case 4: dayString = "Thursday";
            break;
    case 5: dayString = "Friday";
            break;
    case 6: dayString = "Saturday";
            break;
    case 7: dayString = "Sunday";
            break;
    default: dayString = "invalid day";
            break;
}
```

```
switch(month) {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        numberOfDays = 31;
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        numberOfDays = 30;
        break;
    case 2:
        if(isLeapYear())
            numberOfDays = 29;
        else
            numberOfDays = 28;
        break;
}
```

## Βρόχοι

Η Java έχει τρεις βρόχους: *while*, *do-while*, και *for*.

### **while**

Ο βρόχος *while* εκτελεί ένα μπλοκ εντολών για όσο μία δεδομένη παράσταση αποτιμάται σε *true*. Η παράσταση ελέγχεται πριν από την εκτέλεση του σώματος του βρόχου, οπότε το σώμα μπορεί να εκτελεστεί μηδέν φορές (καμία φορά).

```
while (παράσταση) {  
    εντολές  
}
```

Παραδείγματα:

```
int i = 0;  
while(i < text.size()) {  
    System.out.println(text.get(i));  
    i++;  
}  
  
while(iter.hasNext()) {  
    processObject(iter.next());  
}
```

## **do-while**

Ο βρόχος *do-while* εκτελεί ένα μπλοκ εντολών για όσο μία δεδομένη παράσταση αποτιμάται σε *true*. Η παράσταση ελέγχεται μετά από την εκτέλεση του σώματος του βρόχου, επομένως το σώμα εκτελείται πάντα τουλάχιστον μία φορά.

```
do {  
    εντολές  
} while (παράσταση);
```

Παράδειγμα:

```
do {  
    input = readInput();  
    if(input == null) {  
        System.out.println("try again");  
    }  
} while(input == null);
```

## **for**

Ο βρόχος *for* έχει δύο διαφορετικές μορφές. Η πρώτη μορφή είναι επίσης γνωστή ως βρόχος *for-each*, και χρησιμοποιείται αποκλειστικά για τη διάσχιση των στοιχείων μιας συλλογής. Στη μεταβλητή βρόχου ανατίθεται η τιμή των διαδοχικών στοιχείων της συλλογής σε κάθε επανάληψη του βρόχου.

```
for (δήλωση-μεταβλητής : συλλογή) {  
    εντολές  
}
```

Παράδειγμα:

```
for (String note : list) {  
    System.out.println(note);  
}
```

Η δεύτερη μορφή εκτελείται για όσο μία *συνθήκη* αποτιμάται σε *true*. Πριν να ξεκινήσει ο βρόχος, μία εντολή *αρχικοποίησης* εκτελείται μία μόνο φορά. Η *συνθήκη* αποτιμάται πριν από κάθε εκτέλεση του σώματος του βρόχου (οπότε ο βρόχος μπορεί να εκτελεστεί μηδέν φορές). Μία εντολή *επαύξεσης* εκτελείται μετά από κάθε εκτέλεση του σώματος του βρόχου.

```
for (αρχικοποίηση; συνθήκη; επαύξεση) {  
    εντολές  
}
```

Παράδειγμα:

```
for(int i = 0; i < text.size(); i++) {  
    System.out.println(text.get(i));  
}
```